Optical Flow Clustering Using Centroid Neural Network for Motion Tracking of Moving Vehicles

Dong-Min Woo*
Department of Electronics Eng., Myongji University, Yongin, Korea 449-728.

* Corresponding author. Tel.: +82-31-330-6752; email: dmwoo@mju.ac.kr
Manuscript submitted September 18, 2014; accepted May 3, 2015.
doi: 10.17706/jcp.10.3.213-220

Abstract: Motion tracking is one of the most practical applications of computer vision in real life. In this paper, we highlight a new application for tracking motion and estimating the velocity of the moving vehicle in terms of clustering of optical flows. A centroid neural network with a metric utilizing optical flow is employed to group pixels of moving vehicles from traffic images, and to generate blobs of moving vehicles. To verify the best optical flow, we utilize RANSAC (RANdom SAample Consensus) by determining the best model that optimally fits the flows. Experiments are performed with various traffic images. The results show that the proposed method can efficiently segment moving vehicles out of background and accurately estimate the velocity of moving vehicle.

Key words: Tracking, moving vehicle, centroid neural network, clustering.

1. Introduction

Object tracking has been intensely studied for its vast application area, which includes image segmentation, dynamic scene analysis, image registration, visual navigation, motion estimation and video compression [1]. One of the most popular approaches to track motion of object in video frame is to calculate a dense map of optical flows and then do segmenting or clustering on that map. However, most clustering algorithms need an initial number of groups or blobs that is obviously unknown. Other clustering algorithms that do not need the number of groups are so complex and costly to apply.

Optical flow algorithm has aperture problem that has to solve two unknowns in one equation. All optical flow methods try to introduce additional conditions so that we have more constrain to find the optical flow. Lucas-Kanade algorithm [2], a local least square calculation could be sufficient to compute optical flow sparsely for each blob. Another way of determining optical flow is in terms of discrete optimization [3]. The image matching is carried out using label assignment in the quantized search space, and the solution can be optimally found by minimizing the distance. Cross-correlation [4] can be a widely used block-based solution for the determination of optical flow. Sum of squared difference can be also used instead. The pyramid scheme [5] can be used to implement the iterative Lucas-Kanade algorithm for the computation of optical flows of each blob.

In this paper, we are concerned with a grouping process of the optical flows for that incorporates the central neural network (CNN) algorithm. Segmentation of moving vehicles results from this grouping process and motion tracking can be successively carried out.

The CNN is an unsupervised competitive learning algorithm based on the classical k-means algorithm, [6] It has advantageous features such as requiring neither a predetermined schedule for learning coefficients nor the total number of iterations for clustering. Simulation results on clustering problems and image compression problems have shown that CNNs converge much faster than conventional algorithms with a compatible clustering quality while other algorithms including the self-organizing map (SOM) may yield unstable results depending on the initial values of the learning coefficient and the total number of iterations. It is recently reported that the CNN-based grouping method can also be successfully applied to the clustering of line segments for building detection. [7]

With the use of CNN for grouping the optical flows, segmentation of a moving vehicle is straightforward. For each of the segmented blobs of moving vehicles, RANSAC (RANdom SAample Consensus) is employed to verify the best optical flow.

2. Clustering of Optical Flow

2.1. Blob Tracking Procedure

The discussed model is given as follow: The process of blob tracking system is described in Fig. 1. The system analyses frames in a video file and provides blobs information as output.

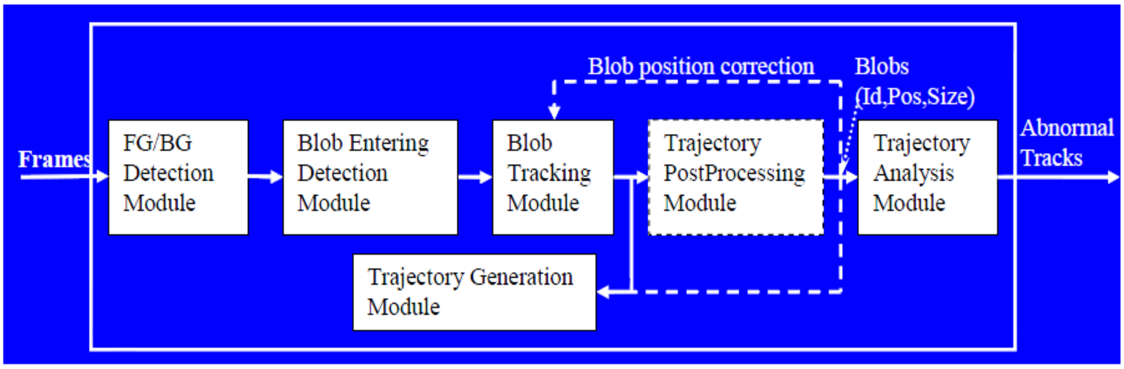

Fig. 1. Structure of blob tracking system.

2.2. Optical Flow Computation

The discussed model is given as follow: The Lucas-Kanade algorithm assumes optical flow is the same for all the pixels in a local neighborhood [2]. This assumption can also be held for moving vehicles. The adjacent pixels of the blob of any moving vehicles should show the same optical flows with respect to *x* and *y* directions, Vx and Vy, respectively, where Vx and Vy are related by the following optical flow equation:

$$\frac{\partial I}{\partial x}Vx + \frac{\partial I}{\partial y}Vy + \frac{\partial I}{\partial t} = 0 \tag{1}$$

In the traffic environment, vehicles are assumed to proceed with the forward direction. In this context, ratio of Vy and Vx is within the small range, and Vx and Vy are consequently related within the limited search space. In this paper, the search space is quantized, and the possible combination of Vx and Vy can be applied to the CNN-based grouping algorithms to find the clusters of the optical flows.

2.3. Centroid Neural Network

The CNN algorithm is an unsupervised competitive learning algorithm based on the classical k-means clustering algorithm. It finds the centroids of clusters at each presentation of the data vector. The CNN first introduces definitions of the winner neuron and the loser neuron. When a data xi is given to the network at the epoch (k), the winner neuron at the epoch (k) is the neuron with the minimum distance to xi. The loser neuron at the epoch (k) to xi is the neuron that was the winner of xi at the epoch (k-1) but is not the winner of xi at the epoch (k). The CNN updates its weights only when the status of the output neuron for the presenting data has changed when compared to the status from the previous epoch.

When an input vector x is presented to the network at epoch n, the weight update equations for winner neuron j and loser neuron i in CNN can be summarized as follows:

$$w_j(n+1) = \frac{1}{N_j+1}\left[N_j w_j(n) + x(n)\right] = w_j(n) + \frac{1}{N_j+1}\left[x(n) - w_j(n)\right] \quad (2)$$

$$w_i(n+1) = \frac{1}{N_i-1}\left[N_i w_i(n) + x(n)\right] = w_i(n) + \frac{1}{N_i-1}\left[x(n) - w_i(n)\right] \quad (3)$$

In Equations (2) and (3), $w_j(n)$ and $w_i(n)$ represent the weight vectors of the winner neuron and the loser neuron, respectively, while N_i and N_j denote the number of data vectors in cluster i^{th} *and* j^{th} at the time of iteration, respectively.

The learning rule for CNN is based on the following theorem and on the condition for minimum energy clustering:

Theorem 1: The centroid of data in a cluster is the solution that gives minimum energy in L_2 norm.

Minimum energy condition: The weights for a given output neuron should be chosen in such a way as to minimize the total distance in L_2 norm from the vectors in its class, such as

$$w_j = \min_w \sum_{i=1}^{N_j} \left\| x_j(i) - w \right\|^2 \quad (4)$$

Using Theorem 1, Equation (4) can be expressed as

$$w_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_j(i) \quad (5)$$

where N_j is the number of members in cluster j.

When CNN is compared with other conventional competitive learning algorithms, the CNN produces very comparable results with less computational effort. That is, the CNN requires neither a predetermined schedule for learning gain nor a total number of iterations for clustering; it converges stably to suboptimal solutions, while the conventional algorithms, including the Self Organizing Map (SOM), may give unstable results depending on the initial learning gain and the total number of iterations.

Other modifications of this clustering technique are realized by increasing the dimensions of the feature space by introducing additional features. By visually comparing the clustering results, the vehicles tracked by using CNN is more stable than that done by FCM and SOM, because the two other algorithms yield different results depending on the initial conditions.

The CNN has several advantages over conventional algorithms such as SOM or k-means algorithm when used for clustering and unsupervised competitive learning. The CNN requires neither a predetermined schedule for learning gain nor the total number of iterations for clustering. It always converges to sub-optimal solutions while conventional algorithms such as SOM may give unstable results depending on the initial learning gains and the total number of iterations. More detailed description on the CNN can be found in literatures.

Note that each cluster obtained after training the CNN represents a certain moving vehicle in the image. The purpose of the clustering algorithm is to link pixels within a moving vehicle to form a blob. Feature space consists of optical flows with respect to x and y directions, Vx and Vy, for which winner neuron and loser neuron are updated by equation (2) and (3). After the CNN algorithm is completely finished, the clusters that represent blobs of moving vehicles in the image are obtained.

3. Motion Tracking and Velocity Estimation

Once blobs are obtained by CNN-based Video Surveillance system, we carry out the processes to have motion direction and velocity of each blob. Fig. 2 is a diagram of this process.

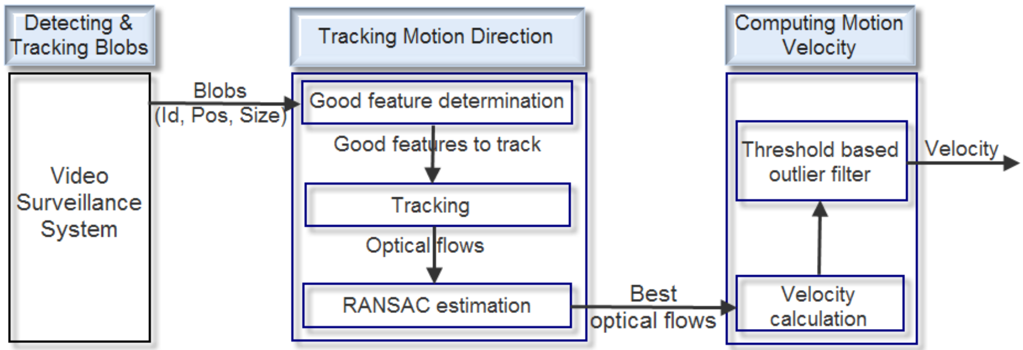

Fig. 2. Process of motion tracking and velocity estimation.

3.1. Blob Tracking

In this first step, we have a series of blobs by using blob tracking system. Instances of this implementation are shown in [8]. We can employ Kalman filtering for the estimation of blob position and size. [9] Each blob will be processed further to determine motion direction as well as motion velocity in next steps.

3.2. Velocity Estimation

RANSAC is used to estimate the best model of data given a small set of inliers and outliers. Once we have the optical flows for each blob, we calculate angle and hypotenuse of each optical flow. Then we use RANSAC to determine the best model that optimally fits the flows. Through the RANSAC, we can eliminate noise from sparse calculating optical flow for 5 features to track.

Using a threshold we find the most appropriate result of velocity. Noises between frames in a static camera's video file can come from many sources like camera's vibration, weather, or simply background noise. Even RANSAC can make mistake when refining from a bad source of optical flows of these noises. In providing a flexible solution to many kinds of video file, a threshold of magnitude of velocity is used to examine the best result from RANSAC process to eliminate outlier. Once the threshold is satisfied, the result is accepted.

4. Experiment

Our application is developed based on Video Surveillance Systems and optical flow tracking system of OpenCV framework [10]. We tested our application with several highway camera video files without any knowledge of the camera parameter. Different video files with a variety of car intensity are used to test flexibility and stability of our application. In Fig. 3, a good quality video file with occlusion effect at separator is examined. Fig. 4 is a scene with much more cars. And in Fig. 5, the traffic video captured in the busy road.

For various occasions, shown in Fig. 3, 4 and 5, our CNN-based clustering algorithm detects each blobs of moving vehicles. Even though some moving vehicles are not detected in an unfavourable situation, most vehicles are successfully tracked by using our clustering algorithms. Considering real world difficulty of tracking vehicles, this initial experimental result is very promising. In Fig. 3, 4 and 5, the detected blobs are denoted by green circles, while the estimated velocity is indicated by red arrow.

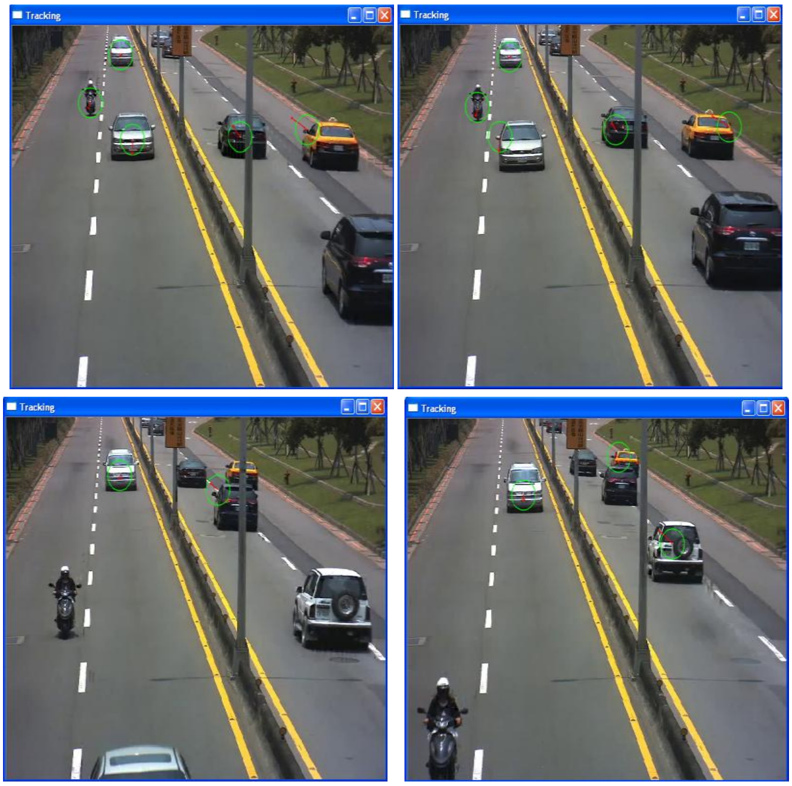

Fig. 3. Motion tracking and velocity estimation with occlusion effect at separator.

Fig. 4. Motion tracking and velocity estimation for many cars.

Fig. 5. Motion tracking and velocity estimation in the busy road.

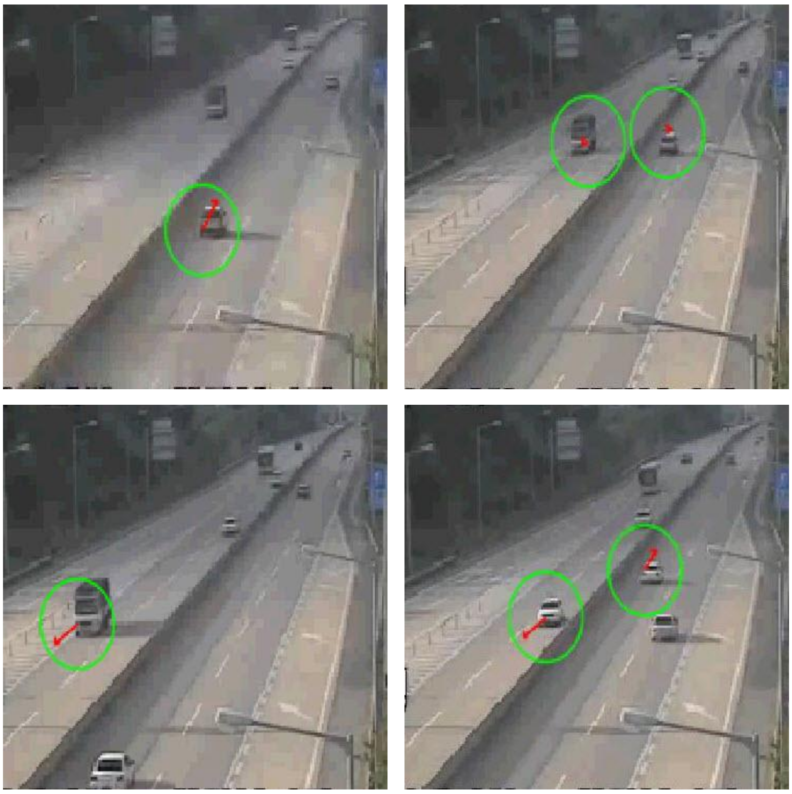

Fig. 6. Motion tracking and velocity estimation in bad quality video file.

Especially in Fig. 6, the image files in the video file are not very clear with low resolution. Moreover, there is a fog, which significantly obscures the tracking procedure. The tracking results show that our CNN-based clustering algorithm detects moving vehicle in this unfavourable situation. Fig. 7 shows the speeds of vehicles tracked in one side of road scene appeared in Fig. 4. The tracking graph indicates how three vehicles in one side of road can be tracked in terms of the clustering of optical flows. In terms of the experimental results, we are convinced that our tracking method can be efficiently applied to the monitoring of traffic information.

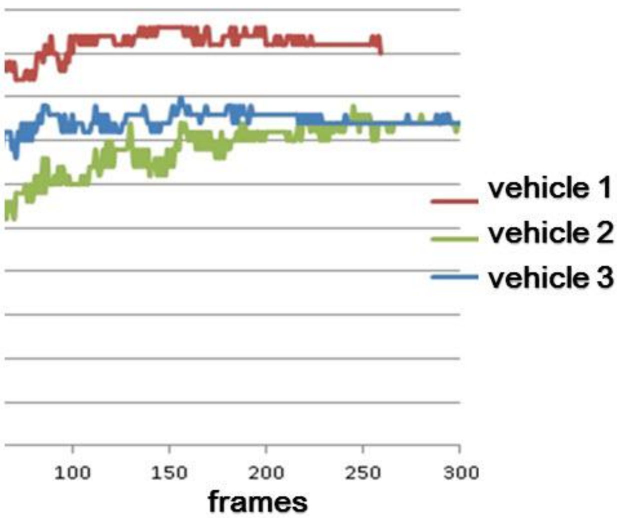

Fig. 7. Motion tracking of three vehicles appeared in Fig. 4.

5. Conclusion

In this paper, we have proposed a new procedure to track and determine velocity of blobs by clustering optical flows. A centroid neural network with a metric utilizing optical flow is employed to group pixels of moving vehicles from traffic images. We also developed an application and test its performance with some different video files of highway traffic camera. The experiments of tracking moving vehicles are carried out from the slight traffic environment to heavy traffic environment. A threshold is applied manually at last to reduce outliers that do not come from real motion in the video. In future research, the proposed method can be improved by including autonomous learning and detection of the value of threshold in order to fit to motion in the video.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (No. NRF-2012R1A1A2004950).

References

[1] Royden, C. S., & Moore, K. D. (2012). Use of speed cues in the detection of moving objects by moving observers. *Vision Research*, *59,* 17-24.

[2] Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging Understanding Workshop* (pp. 121-130).

[3] Glocker, B., Komodakis, N., Tziritas, G., Navab, N. & Paragios, N. (2008). Dense image registration through MRFs and efficient linear programming. *Medical Image Analysis Journal.*

[4] Beauchemin, S. S., & Barron, J. L. (1995). *The Computation of Optical Flow*. ACM New York, USA.

[5] Woo, D.-M., & Hong, A. N. (2014). An integrated solution to motion tracking of moving vehicles using optical flow. *Int. Journal of Computing, Communication and Instrumentation Eng., 1(1)*, 60-63.

[6] Park, D.-C. (2000). Centroid neural network for unsupervised competitive learning. *IEEE Trans. Neural Network,* 520-528.

[7] Park, D.-C., Woo, D.-M., Kim, C.-S. & Min, S.-Y. (2014). Clustering of 3D line segments using centroid neural network for building detection. *Journal of Circuits, Systems and Computers, 23(5).*

[8] Li, L., Huang, W., Gu, I., & Tian, Q. (2003). Foreground object detection from videos containing complex background. *Proceedings of the eleventh ACM international conference on Multimedia* (pp. 2-10).

[9] Zhu, Z., Qiang, J., Fujimura, K., & Lee, K. (2002). Combining Kalman filtering and mean shift for real time eye tracking under active IR illumination. *Proceedings of International Conference on Pattern Recognition* (pp. 318-321).

[10] Stavens, D. (2007). *The OpenCV Library: Computing Optical Flow*, Stanford Artificial Intelligence Lab.

Dong-Min Woo was born in Masan, Korea on Feb. 6, 1958. He got the BS and MS degrees in electronics engineering at Yonsei University, Seoul, Korea, respectively in 1980 and 1982. In 1987, he earned the Ph.D. degree in electrical engineering and applied physics at Case Western Reserve University, located in Cleveland, Ohio, USA. Immediately after finishing Ph.D. degree, we worked for LG Industrial Systems as a senior R&D engineer. Since 1990, he has worked in the Department of Electronics Eng. at Myongji University. His research interest includes computer vision, computational intelligence and remote sensing. Currently, he is a member of KIEE, IEIE and IKEEE.